Practical 1 AI Open File

Implement depth first search algorithm and Breadth First Search algorithm, Use an undirected graph and develop a recursive algorithm for searching all the vertices of a graph or tree data structur

BFS

```
graph = {
    'A' : ['B','C'],
    'B' : ['D', 'E'],
    'C' : ['F'],
    'D' : [],
    'E' : ['F'],
    'F' : []
}
visited = []
queue = []
def bfs(visited, graph, node):
  visited.append(node)
  queue.append(node)
  while queue:
    s = queue.pop(0)
    print (s, end = " ")
    for neighbour in graph[s]:
      if neighbour not in visited:
        visited.append(neighbour)
        queue.append(neighbour)

bfs(visited, graph, 'A')
```

DFS

```
graph = {
    'A' : ['B','C'],
    'B' : ['D', 'E'],
    'C' : ['F'],
    'D' : [],
    'E' : ['F'],
    'F' : []
}

visited = set()
def dfs(visited, graph, node):
  if node not in visited:
    print (node)
    visited.add(node)
    for neighbour in graph[node]:
      dfs(visited, graph, neighbour)

print("Following is the Path using Depth-First Search")
dfs(visited, graph, 'A')
```

Practical 2 AI
Implement A star Algorithm for any game search problem.

```
import heapq

class Node:
    def __init__(self, position, parent=None):
        self.position = position
        self.parent = parent
        self.g = 0
        self.h = 0
        self.f = 0

    def __eq__(self, other):
        return self.position == other.position

    def __lt__(self, other):
        return self.f < other.f

def astar(grid, start, goal):
    open_list = []
    closed_list = set()
    start_node = Node(start)
    goal_node = Node(goal)
    heapq.heappush(open_list, (0, start_node))
    while open_list:
        current_node = heapq.heappop(open_list)[1]
        closed_list.add(current_node.position)
        if current_node == goal_node:
            path = []
            while current_node is not None:
                path.append(current_node.position)
                current_node = current_node.parent
            return path[::-1]
        children = []
        for new_position in [(0, -1), (0, 1), (-1, 0), (1, 0)]:
            node_position = (current_node.position[0] + new_position[0],
                             current_node.position[1] + new_position[1])
            if (0 <= node_position[0] < len(grid)) and \
               (0 <= node_position[1] < len(grid[0])) and \
               (grid[node_position[0]][node_position[1]] == 0) and \
               (node_position not in closed_list):
                new_node = Node(node_position, current_node)
                children.append(new_node)
        for child in children:
            child.g = current_node.g + 1
            child.h = ((child.position[0] - goal_node.position[0]) ** 2) + \
                      ((child.position[1] - goal_node.position[1]) ** 2)
            child.f = child.g + child.h
            for open_node in open_list:
                if child == open_node[1] and child.g > open_node[1].g:
                    continue
            heapq.heappush(open_list, (child.f, child))
    return None
grid = [
    [0, 0, 0, 0, 0],
    [0, 1, 1, 1, 0],
    [0, 0, 0, 0, 0],
    [0, 1, 1, 1, 0],
    [0, 0, 0, 0, 0]
]
start = (0, 0)
goal = (4, 4)

path = astar(grid, start, goal)
if path:
    print("Path found:", path)
else:
    print("No path found.")
```

Practical 3 AI Open File

Implement Greedy search algorithm for Selection Sort

```
x=[]
n=int(input("Enter how many no you want for Selection Sorting "))
for i in range(n):
 print("Enter the Element",i,": ")
 a=int(input(""))
 x.append(a)

print("unsorted element are", x)
for i in range(0,len(x)-1):
 for j in range(i+1,len(x)):
  if x[i]>x[j]:
   c=x[i]
   x[i]=x[j]
   x[j]=c
```

Practical 4 AI Open File

Implement a solution for a Constraint Satisfaction Problem using Branch and Bound and Backtracking for n-queens problem or a graph coloring problem.

```
global N
N = 4
def printSolution(board):
  for i in range(N):
    for j in range(N):
      print (board[i][j],end=' ')
    print()

def isSafe(board, row, col):
  for i in range(col):
    if board[row][i] == 1:
      return False
  for i, j in zip(range(row, -1, -1), range(col, -1, -1)):
    if board[i][j] == 1:
      return False
  for i, j in zip(range(row, N, 1), range(col, -1, -1)):
    if board[i][j] == 1:
      return False
  return True

def solveNQUtil(board, col):
  if col >= N:
    return True
  for i in range(N):
    if isSafe(board, i, col):
      board[i][col] = 1
      if solveNQUtil(board, col + 1) == True:
        return True
      board[i][col] = 0
  return False

def solveNQ():
  board = [ [0, 0, 0, 0],
          [0, 0, 0, 0],
           [0, 0, 0, 0],
           [0, 0, 0, 0]
         ]
  if solveNQUtil(board, 0) == False:
    print ("Solution does not exist")
    return False
  printSolution(board)
  return True

solveNQ()
```

Practical 5 AI Open File

Develop an elementary chatbot for any suitable customer interaction application

```
def greet(bot_name,birth_year):
  print("Hello, my name is ",bot_name, format(bot_name))
  print("I was created in {0}", format(birth_year))

def remind_name():
  print("Please, remind me your name")
  name=input()
  print("What a great name you have, ",name,"!", format(remind_name))

def guess_age():
  print("Let me guess your age")
  print("Enter remainder of dividing your age by 3,5 and 7")
  rem3=int(input())
  rem5=int(input())
  rem7=int(input())
  age=(rem3*70+rem5*21+rem7*15)%105
  print("Your age is",age,", that's great time to start programming!")

def count():
  print("How I will prove to you that I can count to any number you want")
  num=int(input())
  count=0
  while count<=num:
    print("{0}!", format(count))
    count+=1

def test():
  print("let test your program knowldege")
  print("Why we do you use method?")
  print("1. To repeat a statement in multiple time")
  print("2. To decompose a program into sevaral small subroutine")
  print("3. To determine the execution time of a program")
  print("4. To interupt the execution of a program")
  answer=2
  guess=int(input())
  while guess!=answer:
    print("Please, try again")
    guess=int(input())
  print("Complete, have a nice day")

def end():
  print("Congratulations, you won")

greet('TE-Chatbot','2024')
remind_name()
guess_age()
count()
test()
end()
```

Practical 6 AI Open File

Implement Help desks management System

```
problem_dict = {
   "Printer not working": "Check that it's turned on and connected to the network",
   "Can't log in": "Make sure you're using the correct username and password",
   "Software not installing": "Check that your computer meets the system requirements",
   "Internet connection not working": "Restart your modem or router",
   "Email not sending": "Check that you're using the correct email server settings"
}

def handle_request(user_input):
  if user_input.lower() == "exit":
    return "Goodbye!"
  elif user_input in problem_dict:
    return problem_dict[user_input]
  else:
    return "I'm sorry, I don't know how to help with that problem."

while True:
  user_input = input("What's the problem? Type 'exit' to quit. ")
  response = handle_request(user_input)
  print(response)
```

Practical 7 CC Open File

Installation and configure Google App Engine.

Step 1:- Search Google Cloud Platform in a any search engine& Click onConsole.

Step 2:- Click on select new project.

Step3:- Give Project name and click on create.

Step4:- Click on select project

Step5:- In a search bar type search App Engine

Step 6:-Click on App Engine and following screen will appear& Click onCreate Application.

Step 7:- Click on next (error Occurs ignore that)

Step 8:- Scroll down and click on I'll do this later.(skip this step if error occurs in step 7)

Step 9:- In search bar type App Engine Admin API.

Step 10:- Click Enables

Step 11:- Click Activate Cloud Shell:-

Step 12 :- Following screen will appear:-

Step 13:- Login into your GitHub account and click on new repository.

Step 14 :- Give name to your repository and click create.

Step 15:- Click on creating new file

Step 16 :- Give name to the python file & type your code. print("Hello World")

Step 17:- Click on code and copy URL.(copy HTTPS url from code dropdown menu)

Step 18:- Go to cloud and type – git clone and paste url.

Step 19 :- Type ls

Step 20 :- Enter cd-repository name.

Step 21:- Type ls and to run python code type python-program name.

Practical 8 CC Open File
Creating an Application in SalesForce.com using Apex programming Language

Step No 1: Create new org:
https://developer.salesforce.com/signup

Step No 2: After signup, loging using following URL
https://login.salesforce.com/

Step No 3: Login Page (Enter your credential to login)
1. Open Developer Console (in Setting Logo)
2. File ---> New ---> Select Apex Class Type below mentioned code

```
public class firstClass1 {
        public static void Addition(){
                Integer a = 4; Integer b = 5; Integer c = a + b; Integer d = 4 + 5; Integer e = a + 5;
                System.debug('Add = ' + c); System.debug('Add =' + d); System.debug('Add =' + e);
        }
        public static void Subtraction(){
                Integer a = 4; Integer b = 5; Integer c1 = a - b;
                Integer d1 = b - a; Integer e1 = 4 - 5; Integer f1 = a - 5;
                System.debug('Sub ='+ c1); System.debug('Sub =' + d1); System.debug('Sub =' + e1); System.debug('Sub =' + f1);
        }
        public static void Multi(){
                Integer a = 4;
                Integer b = 5;
                Integer c = a * b; Integer d = 4 * 5; Integer e = a * 5;
                System.debug(c); System.debug(d); System.debug(e);
                Integer f = -4; Integer g = a * f; System.debug(g);
        }
        public static void Div(){
                Integer a = 4;
                Integer b = 5; Integer c = a / b; Integer d = 4 / 5; Integer e = a / 5;
                System.debug(c); System.debug(d); System.debug(e);
        }
}
```

3. Click on Debug ---> Open Execute Acronymous Window
4. Type below code(Apex Code)

```
firstClass1.Addition();
firstClass1.Subtaction();
firstClass1.Multi();
firstClass1.Div();
```

5. Click on Open log then Execute code
6. Click on Debug only (You will get output)

Practical 9 CC